

LAPORAN PRAKTIKUM STRUKTUR DATA
ARRAYLIST PADA JAVA



DOSEN PENGAMPU :

Dr. Wahyudi, M.T.

OLEH :

RIFKI YULIANDRA

NIM.2311532011

DEPARTEMEN INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS ANDALAS
PADANG
2024

Arraylist pada Java

I. TUJUAN

- 1.1. Memahami apa yang dimaksud dengan ArrayList
- 1.2. Memahami cara kerja penambahan data di ArrayList
- 1.3. Memahami cara kerja akses data di ArrayList
- 1.4. Memahami cara kerja penghapusan data di ArrayList.

II. TEORI

Dalam pengembangan Java, mengelola kumpulan elemen dengan tipe data yang beragam adalah tugas yang sering dilakukan. Meskipun array tradisional menyediakan dasar untuk tujuan ini, mereka memiliki batasan ukuran tetap saat dibuat. Keterbatasan ini mengharuskan dibuatnya array baru secara keseluruhan saat volume data berubah.

Kelas `ArrayList`, yang berada di dalam Java Collections Framework, mengatasi tantangan ini. Ia menawarkan struktur seperti array yang dinamis yang dapat beradaptasi dengan lancar terhadap perubahan kebutuhan data. Tidak seperti array statis, `ArrayList` memungkinkan developer untuk menambah atau menghapus elemen sesuai kebutuhan, sehingga tidak perlu menentukan ukuran koleksi sebelumnya. Sifat dinamis ini menjadikan `ArrayList` sebagai landasan untuk mengelola koleksi di mana volume data tidak pasti pada saat program dimulai.

Dalam bahasa pemrograman Java, ArrayList adalah sebuah Class array yang ditemukan dalam paket `java.util`, khususnya dalam `java.util.ArrayList`; Interface `List` digunakan untuk mengimplementasikan class ini. Fungsi ArrayList sangat dinamis, dengan ukuran array tidak terbatas. Dengan demikian, dibandingkan dengan array built-in Java, kita dapat membuat ArrayList dengan ukuran yang lebih fleksibel sesuai dengan isi yang kita masukkan. Akibatnya, ArrayList memiliki banyak fitur tambahan yang tidak dapat diakses pada array biasa.

Secara umum, ArrayList memiliki sintaks deklarasi seperti berikut:
ArrayList<tipe_data> nama_variabel = new ArrayList<>(parameter);

Beberapa fitur penting pada ArrayList di Java adalah sebagai berikut:

1. ArrayList mewarisi Class AbstractList dan mengimplementasi List Interface
2. ArrayList dapat dideklarasikan dengan sebuah ukuran. Namun, ukuran tersebut dapat bertambah secara otomatis jika list pada ArrayList bertambah atau berkurang jika sebuah objek pada list tersebut berubah
3. ArrayList mampu diakses secara acak
4. ArrayList tidak dapat dideklarasikan dengan tipe data primitif seperti int, char, dan sebagainya. Maka dari itu, saat pendeklarasiannya, kita gunakan wrapper class. Contohnya, jika kita ingin menggunakan tipe data int, kita bisa menggunakan sintaks Integer
5. ArrayList di Java dapat dilihat sebagai vektor di C++

Seperti yang telah dijelaskan di atas, perbedaan utama antara Array dan ArrayList adalah ukuran sistem. Jika kita ingin memuat variabel untuk menampung sejumlah data tipe data yang sama, Array berfungsi. Namun, kita harus menentukan ukuran Array, dan Array tersebut terbatas pada ukurannya. Jika sudah dibuat, ukuran tersebut tidak akan pernah berubah.

Pada ArrayList, ukurannya sudah ditentukan pada awal deklarasi, tetapi ukurannya dapat berubah tergantung pada banyaknya data yang ditambahkan nanti. Selain itu, ArrayList hanya dapat menyimpan objek. Ini berarti bahwa mereka tidak dapat mendeklarasikan tipe data primitif tanpa wrapper class. Contohnya adalah mengubah int menjadi Integer. Ini juga dapat bermanfaat karena kita dapat menyimpan object-object yang telah kita buat menjadi ArrayList.

Pada praktikum kali ini, kita akan memahami bagaimana cara kerja operasi yang terdapat pada ArrayList, seperti penambahan data, cara akses data, dan menghilangkan data.

III. LANGKAH KERJA PRAKTIKUM

3.1. Akses indeks ArrayList



```
package pekan_1;

public class arrayReview1 {

    public static void main(String[] args) {
        //declares an Array of Integers.
        int[] arr;
        // allocating memory for 5 integers.
        arr = new int [5];

        // initialize the first element of the array
        arr[0] = 10;

        //initialize the second element of the array
        arr[1] = 20;
        //so on...
        arr[2] = 30;
        arr[3] = 40;
        arr[4] = 50;
        // accessing the elements of the specified array
        for (int i = 0; i < arr.length; i++) {
            System.out.println("Element at index " + i + " : " + arr[i]);
        }
    }
}
```

Dalam arraylist, kita dapat mengakses sebuah index dengan cara:
Nama_array[index ke-n];

Cara akses ini hampir mirip seperti built-in array yang terdapat pada Java. Bisa dilihat pada program tersebut, kita mengalokasikan memory sebanyak 5, dan memasukkan nilai masing masing indeks yaitu 10,20,30,40,dan 50. Lalu, masuk ke bagian perulangan dimana setiap ke-i akan mengambil nilai indeks ke-i.

Contoh hasil keluarannya adalah sebagai berikut:

```
<terminated> arrayReview1 [Java Application]
Element at index 0 : 10
Element at index 1 : 20
Element at index 2 : 30
Element at index 3 : 40
Element at index 4 : 50
```

3.2. Memasukkan elemen ke ArrayList

```
package pekan_1;

import java.util.ArrayList;

public class ArrayListGetElemen {

    public static void main(String[] args) {
        ArrayList<Integer> list = new ArrayList<Integer>();
        // add the number
        list.add(9);
        list.add(5);
        list.add(6);
        System.out.println(list);
        // get method
        Integer n = list.get(1);
        System.out.println("Pada Indeks ke 1 angkanya adalah: " + n);
    }
}
```

Pada program tersebut, kita bisa memasukkan nilai/elemen ke dalam array dengan method:

Nama_array.add(value);

Lalu, berbeda dengan program sebelumnya, kita juga bisa mengambil nilai dari index array dengan method:

Nama_array.get(index ke-i);

Pada method tersebut program akan menggunakan metode get untuk mengambil elemen pada indeks ke-n (pada contoh mengambil elemen pada indeks ke-1) dari list dan menyimpannya ke variable n.

Contoh hasil keluarannya adalah sebagai berikut:

```
<terminated> ArrayListGetElemen [Java Application] C:\U
[9, 5, 6]
Pada Indeks ke 1 angkanya adalah: 5
```

3.3. Menyisipkan data

```
package pekan_1;

import java.util.ArrayList;

public class ArrayListSisip {

    public static void main(String[] args) {
        ArrayList<Integer> list = new ArrayList<Integer>();
        list.add(1);
        list.add(2);
        list.add(4);
        System.out.println(list);
        // Insert missing element 3
        list.add(2, 3);
        System.out.println(list);
    }
}
```

Pada program tersebut, kita bisa menyisipkan sebuah elemen ke index ke-n dengan cara:

Nama_array.add(index ke-n, value);

Method tersebut akan menyisipkan sebuah value kedalam index ke-n. Jika didalam index tersebut sudah ada value, value tersebut akan dipindahkan ke index ke-n+1.

Dari program tersebut, diberikan sebuah susunan angka 1,2,dan 4. Kita ingin menyisipkan angka 3 pada index ke-2 (dalam susunan tersebut berarti baris ke-3). Maka, angka 4 akan dipindahkan ke index ke-3. Bisa dilihat bahwa pengalokasian ukuran pada ArrayList sangat fleksibel dan ukuran tersebut bisa berubah tergantung banyaknya elemen yang kita berikan.

Contoh keluarannya adalah sebagai berikut.

```
<terminated> ArrayListSisip
[1, 2, 4]
[1, 2, 3, 4]
```

3.4. Penghapusan elemen

```
package pekan_1;

import java.util.ArrayList;

public class ArrayListIterasi {

    public static void main(String[] args) {
        // Creating an ArrayList of string type
        ArrayList<String> al = new ArrayList<>();
        // adding elements to ArrayList
        // using standart add() method
        al.add("saye");
        al.add("informatika");
        al.add(1, "mahasiswa");
        // Using the Get method and the for loop
        for (int i = 0; i < al.size(); i++) {
            System.out.println(al.get(i) + " ");
        }
        System.out.println();

        // Using the for each loop
        for (String str : al) {
            System.out.print(str + " ");
        }
        System.out.println();

        // Printing all elements of ArrayList
        System.out.println("Initial ArrayList " + al);
        // Removing element from above ArrayList
        al.remove(1);

        // Printing the updated ArrayList elements
        System.out.println("After the Index Removal " + al);
        // Removing this word element in ArrayList
        al.remove("saye");

        // Now printing updated ArrayList
        System.out.println("After the Object Removal " + al);
    }
}
```

Pada program tersebut, kita bisa menghapus sebuah index atau value dengan method:

Nama_array.remove(index ke-n);

Atau

Nama_array.remove(value);

Kedua tersebut bisa dijalankan sesuai kebutuhan. Kita bisa menghapus sebuah elemen jika ahnya diketahui valuenya saja atau pada index ke berapa value tersebut berada.

Contoh keluarannya adalah sebagai berikut:

```
<terminated> ArrayListIterasi [Java Application] C:\Users\Userr\p2\pool\plug  
saya  
mahasiswa  
informatika  
  
saya mahasiswa informatika  
Intial ArrayList [saya, mahasiswa, informatika]  
After the Index Removal [saya, informatika]  
After the Object Removal [informatika]
```

IV. KESIMPULAN

Berdasarkan praktikum yang penulis lakukan, kita dapat mengetahui bahwa ArrayList dalam bahasa pemrograman Java adalah sebuah Class array yang tersedia dalam paket java.util, tepatnya dalam java.util.ArrayList. ArrayList memiliki ukuran yang dapat disesuaikan dan memungkinkan programmer untuk menyimpan elemen pada satu variabel.